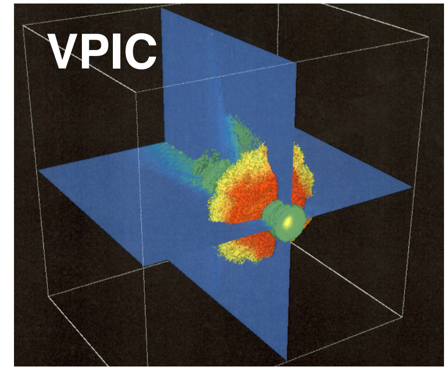
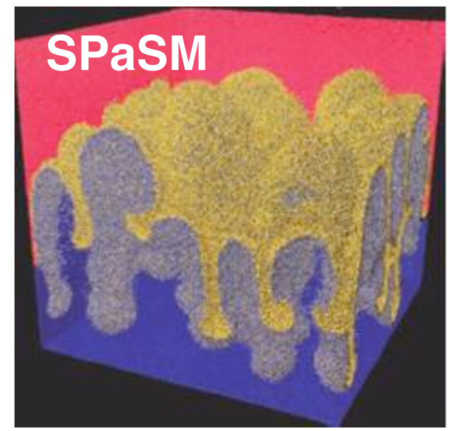
LA-UR-08-06593



Application Design Considerations for Roadrunner and Beyond

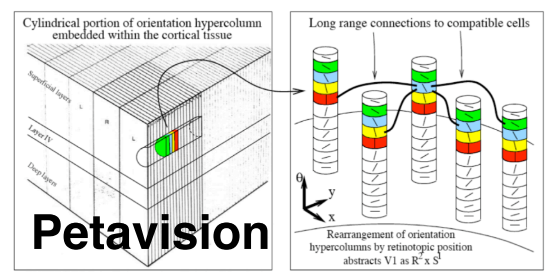


Brian J. Albright

Applied Physics Division, LANL

Los Alamos Computer Science Symposium

Oct 14, 2008



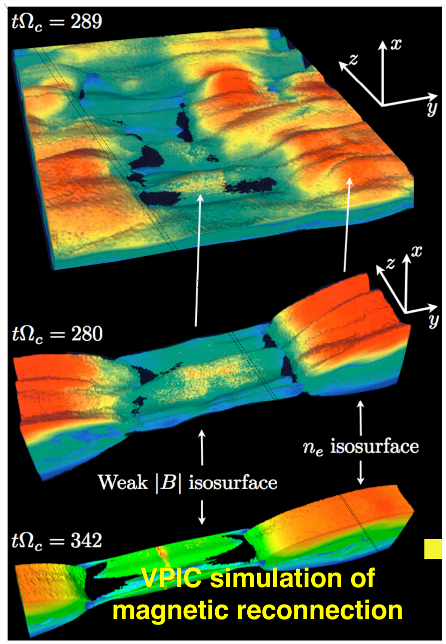
Acknowledgments

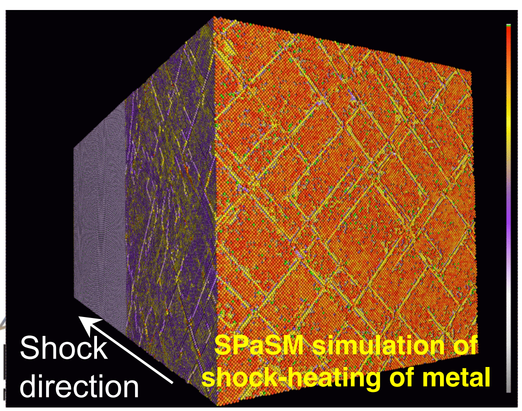
- Kevin Bowers, Ben Bergen, Lin Yin, Thomas Kwan, Charlie Snell, K. Barker, D. Kerbyson, J. Turner, S. Swaminarayan, Tim Germann, Paul Henning, Tim Kelley, Ken Koch, Mike Lang, Jamaludin Mohd-Yusof, Scott Pakin
- IBM
- ASC, LDRD

Outline

- Trends in supercomputing and opportunities for science
- Changes in approach to programming on these platforms
- Roadrunner
- How Roadrunner exposes what one must do to use platforms effectively
- Case study: VPIC design and how we evolved to use the architecture
- Performance and outlook

In the next 10 years, rapid increase in computing power will change the science landscape

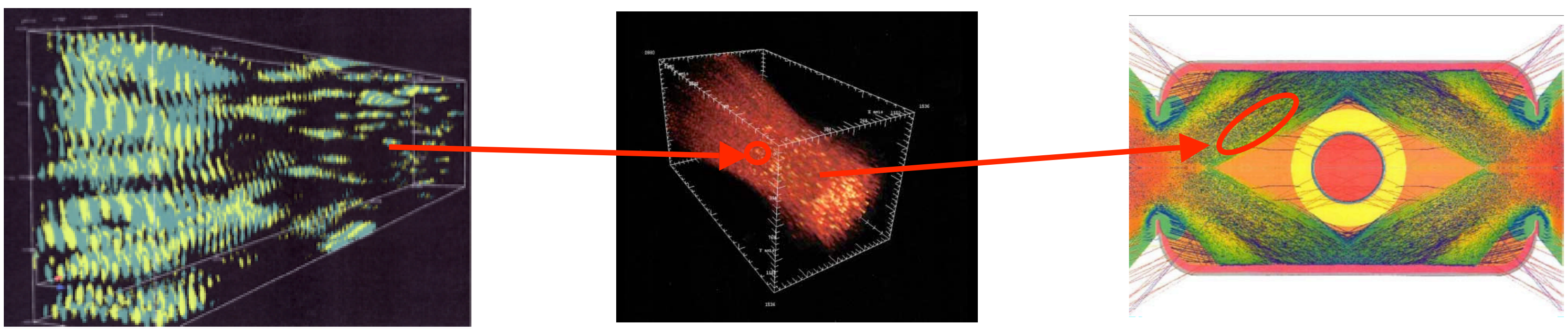




- Petaflop/s computing is here today
- In ten years, we'll have Exaflop/s
- With a few exceptions, experimental or observational facilities will <u>not</u> see a comparable increase in fidelity/size/scale.
- Many if not most of the major discoveries in the next decade will be fueled by computation
  - Plasma and high-energy-density science: "at scale" kinetic modeling of many decades-old problems
  - Materials modeling: full-grain and multi-grain ab initio modeling
  - Predictive climate modeling
  - Computational cosmology
  - Protein folding and computational drug design
  - Modeling of cognition

Another example: risk mitigation for ICF ignition experiments on the National Ignition Facility

- In 2010, fusion ignition experiments start on the multi-billion dollar NIF. The biggest source of uncertainty is whether laser-plasma instabilities (LPI) will prevent ignition. (See *JASON Review Report JSR-05-340*, Section 1.3 Critical Recommendations)
- Petascale supercomputing will help answer these questions.

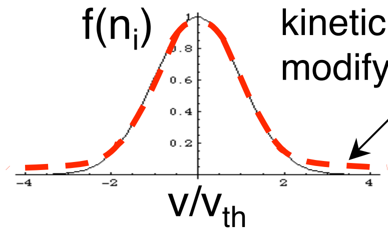


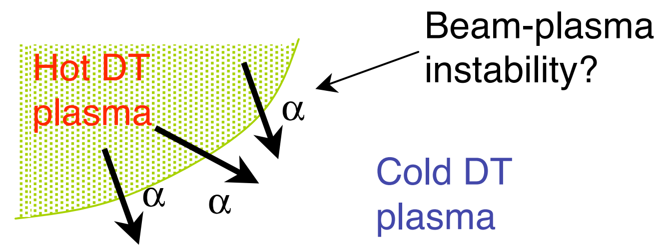
VPIC modeling of a single laser speckle | **LLNL pF3D modeling of a laser beam** | **Integrated LLNL Hydra modeling of ICF experiment**

(Yin et al. PRL 2007; Bowers et al. ACM/IEEE Supercomputing 08 Gordon Bell Prize paper).

Another example: ab initio modeling can change our basic understanding of thermonuclear burn

Kinetic & collective physics can affect TN burn





<u>The challenge for modeling</u>: span the large separation in length and time scales:

$\omega_{pe} \sim 3 \times 10^8$, $\omega_{pi} \sim 4 \times 10^6$, $\nu_{\alpha e} \sim 60$, $\nu_{\alpha I} \sim 3$, $\nu_{DT} \sim 1.3$ (ns^{-1}, NIF-relevant regime)

Collective & kinetic effects may supercede binary collisions

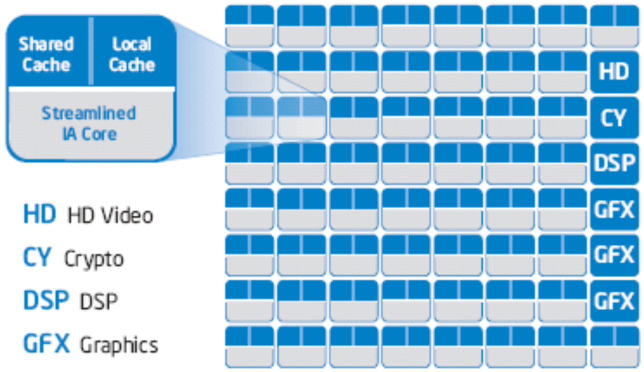
- **Large α population may excite beam-plasma type instability**
  Can change e-i split of α energy
- **Non-maxwellian ions in Gamov peak can change $\langle\sigma v\rangle$**
- **Magnetic fields reduce electron heat conduction (ICF)**

Separation of time scales requires long, large-scale simulations
$\Rightarrow$ Cells, PF-scale machines

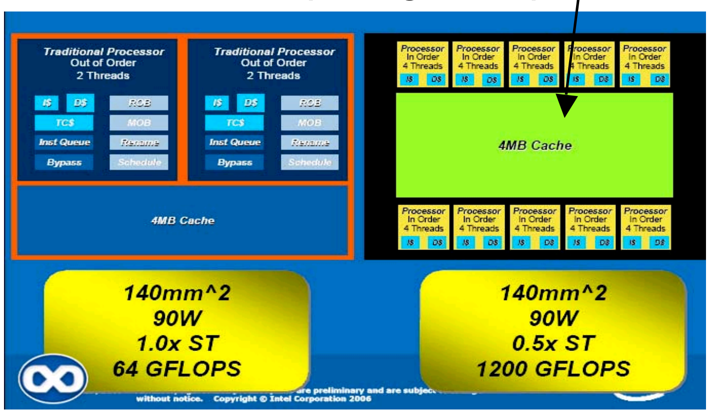
Caveat: Tomorrow's supercomputers probably won't look like today's

Processors are evolving toward hybrid, asymmetric mixes of general and special purpose

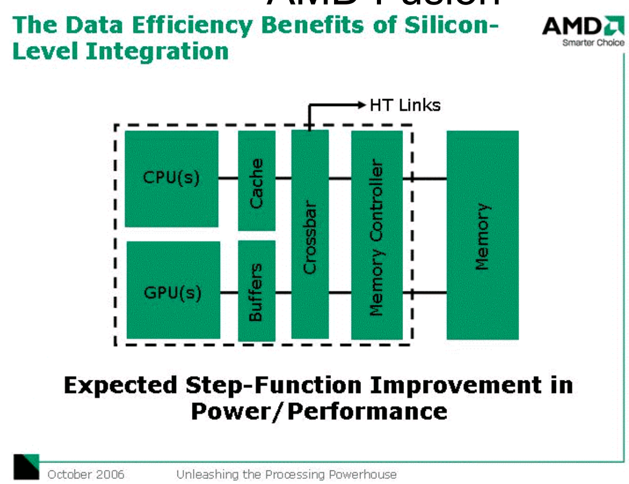
Intel's Microprocessor Research Lab



Intel's Visual Computing Group - Larabee



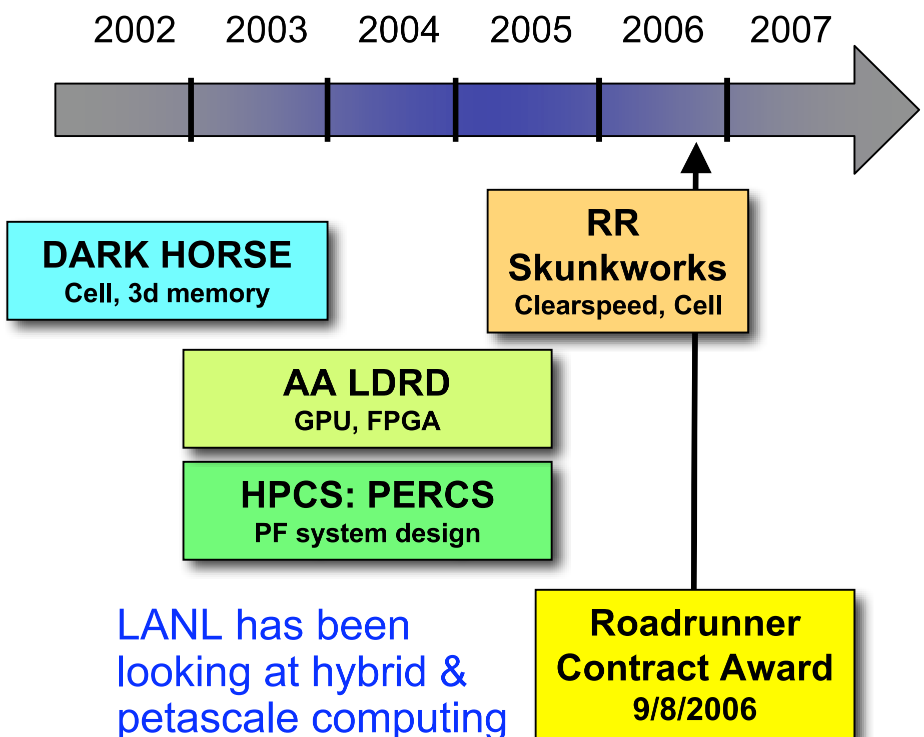
AMD Fusion

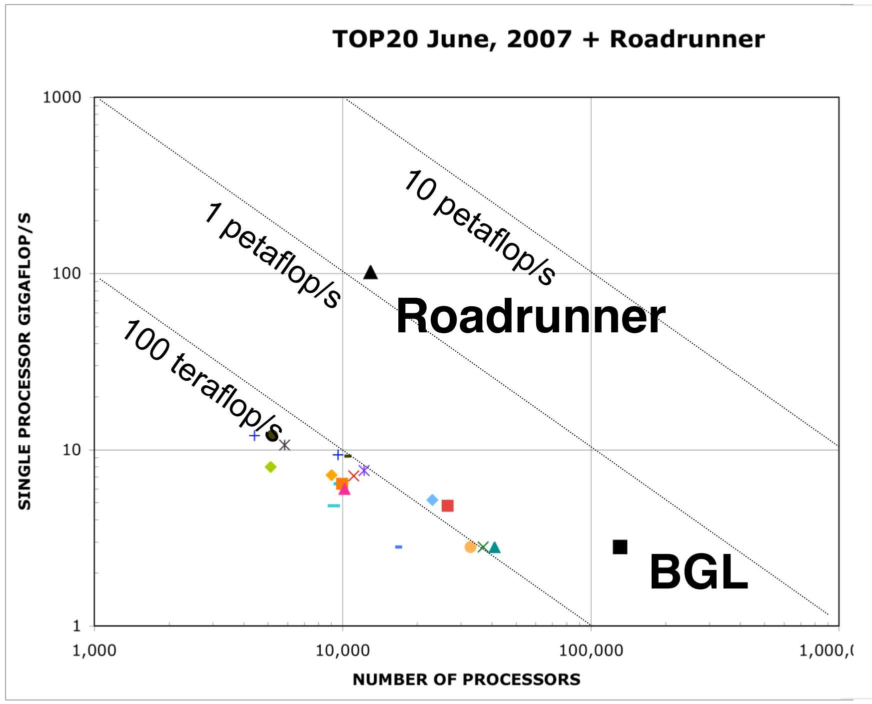


nVidia G80 - 2006

Taken from publicly available information

Hybrid computing is a transformational technology





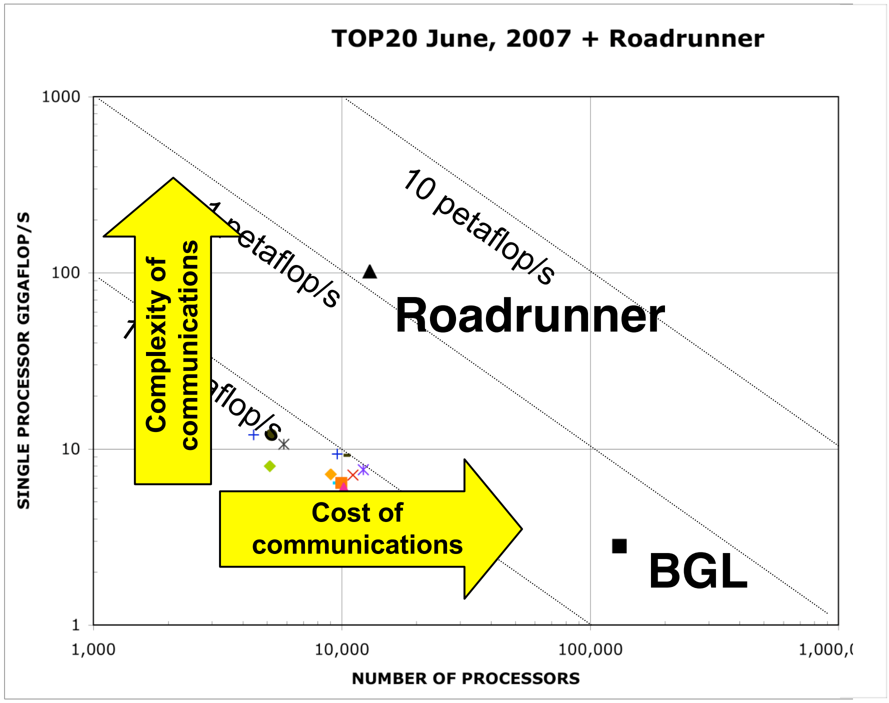
LANL has been looking at hybrid & petascale computing for some time

Roadrunner is a different path to a petascale system

To applications programmers, each axis confers its own challenges

- Vertical axis: increased complexity
  - Deep memory hierarchies
  - Potentially limited localstore (e.g. 256k for Cell SPE)
  - Different instruction sets for accelerator chips
  - Tools are evolving to hide some of this complexity

- Horizontal axis: increased cost
  - Will today's apps that work fine on up to ~100k MPI ranks scale to billion-way parallelism (as required for Exaflop/s computing under the BGL model)?

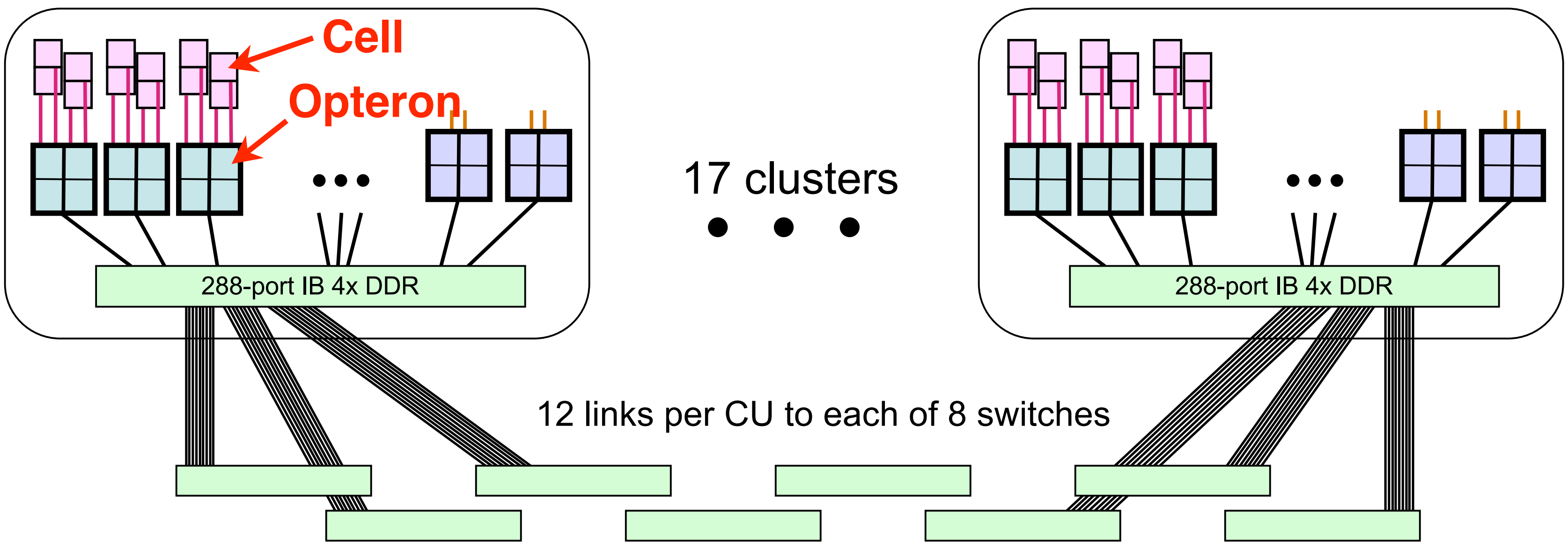


Roadrunner exposes design concepts for achieving high performance on modern architectures

Roadrunner is a cluster of clusters of Cell-accelerated Opteron chips

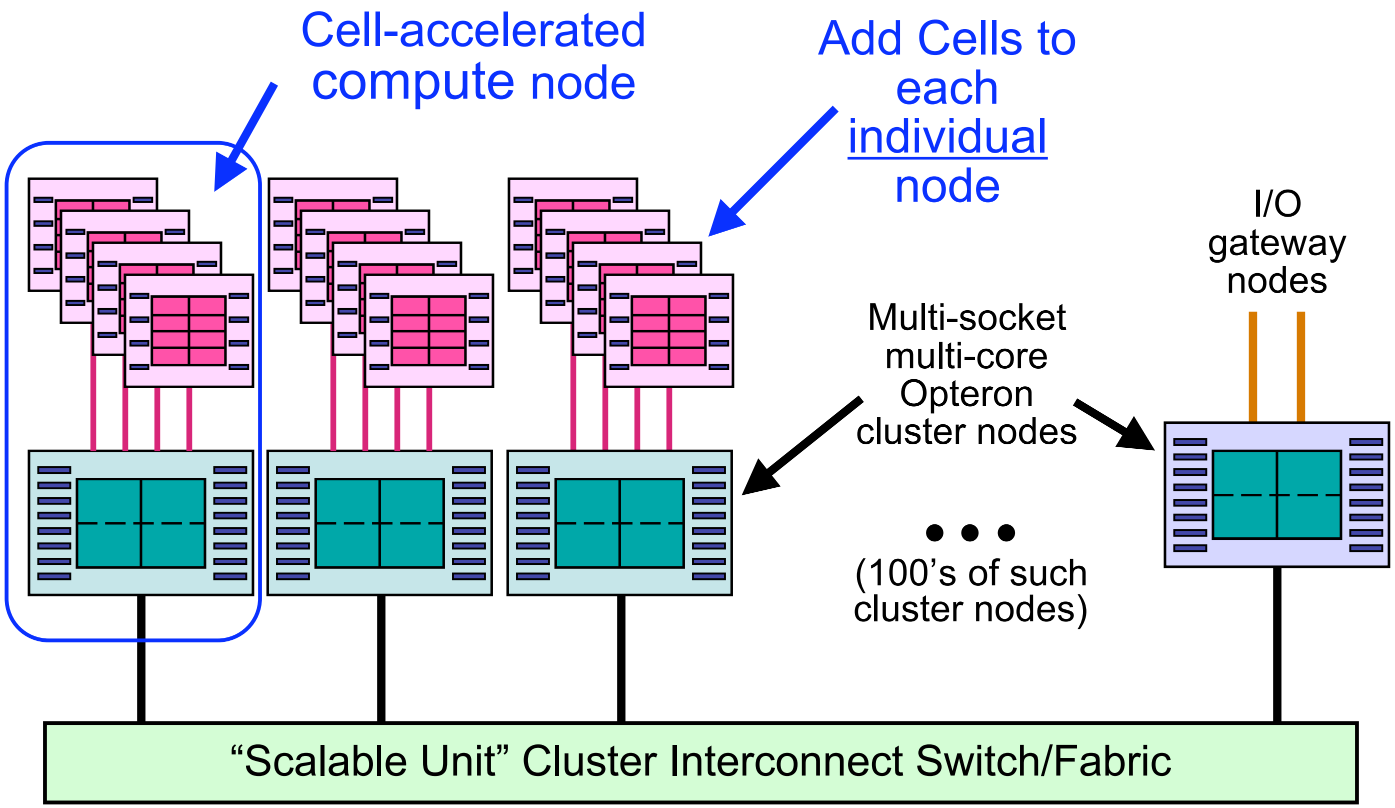
Connected Unit cluster
180 Triblade compute nodes w/ Cells
12 I/O nodes

6,120 dual-core Opterons ⇒ 22.0 Tflop/s (DP)
12,240 Cell eDP chips ⇒ 1.3 Pflop/s (DP)



Eight 2nd-stage 288-port IB 4X DDR switches

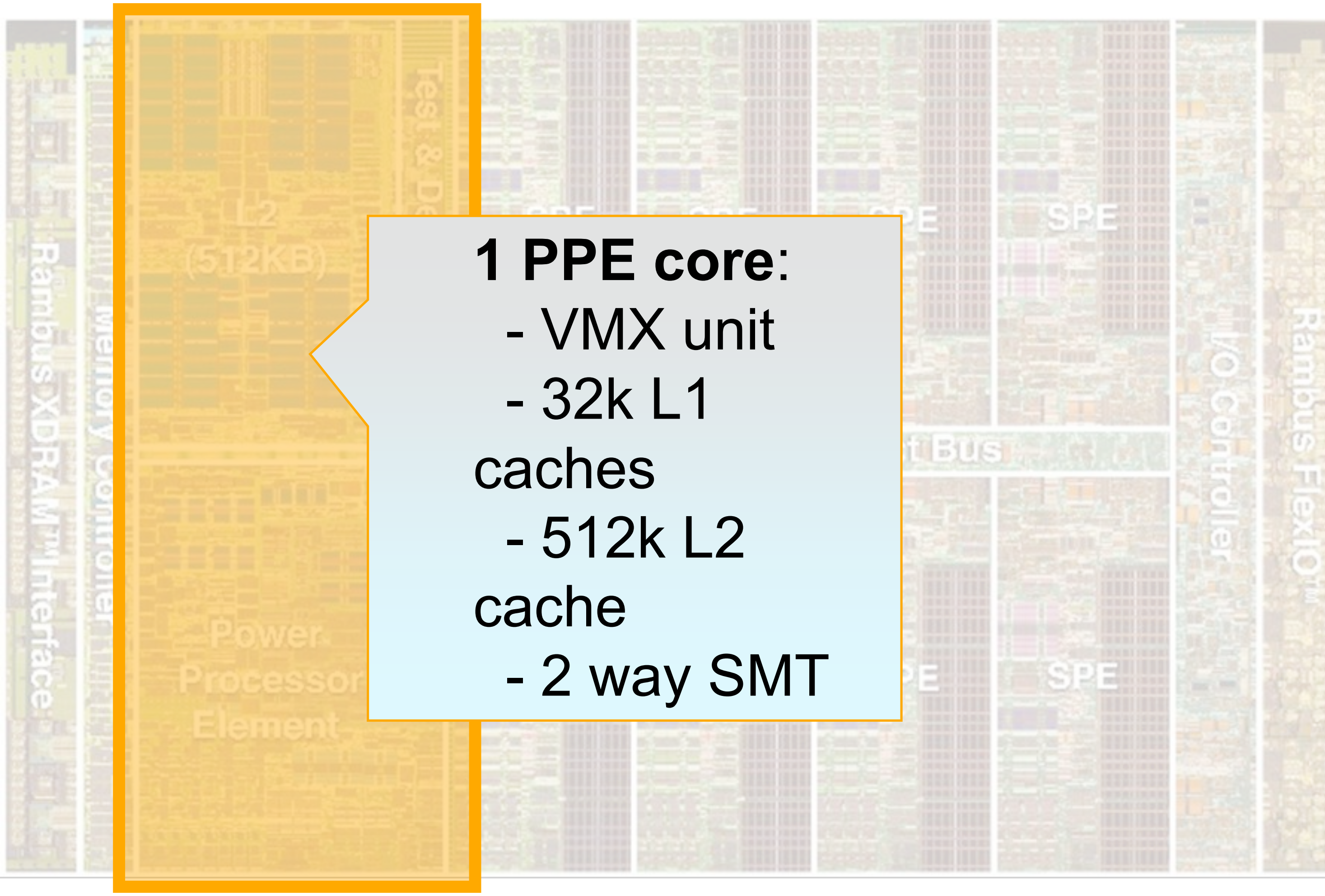
Roadrunner is Cell-accelerated, not a cluster of Cells



Node-attached Cells is what makes Roadrunner different!

Cell Broadband Engine - quick anatomy lesson

Power Processing Element



1 PPE core:

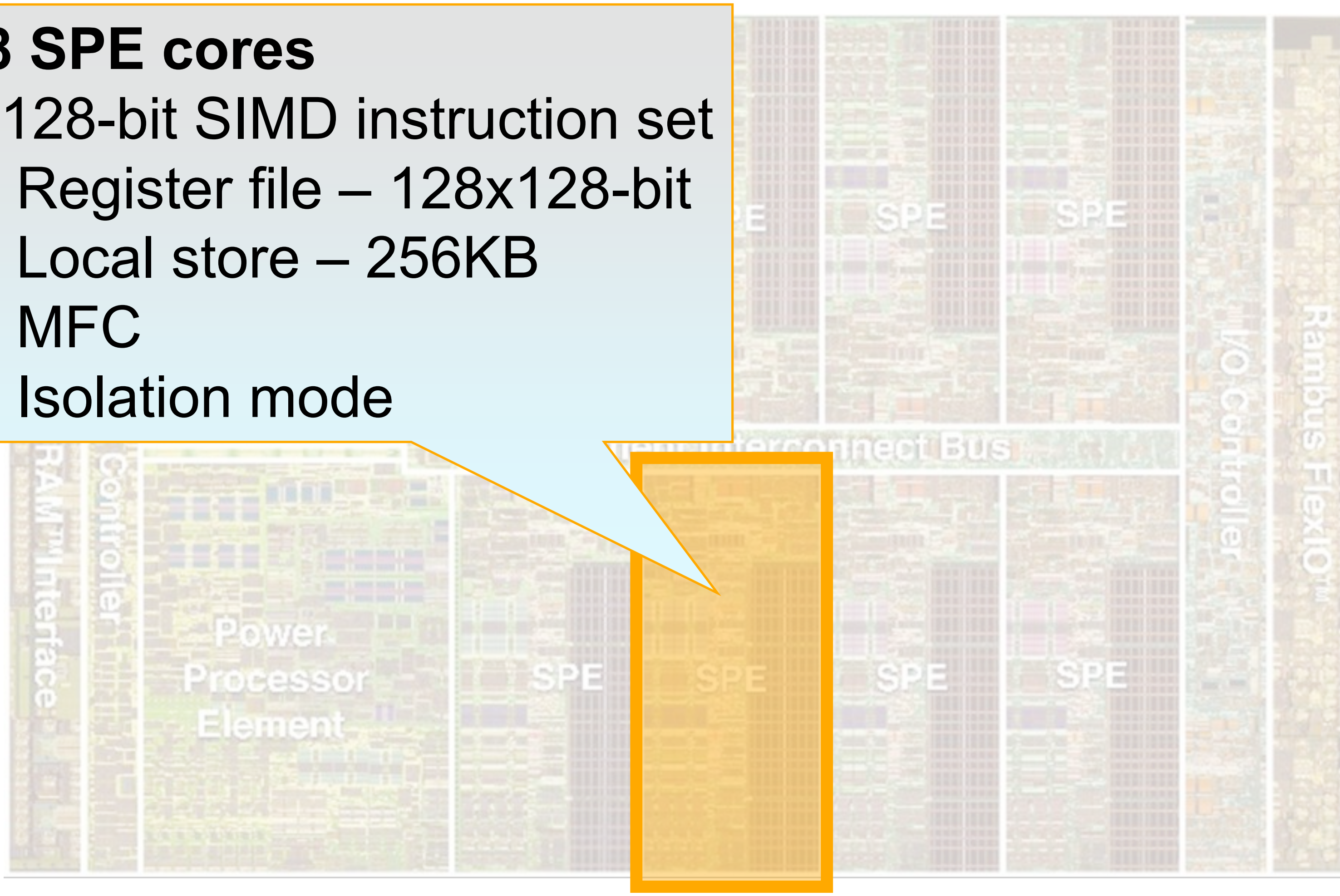
- VMX unit
- 32k L1 caches
- 512k L2 cache
- 2 way SMT

8 Synergistic Processing Elements

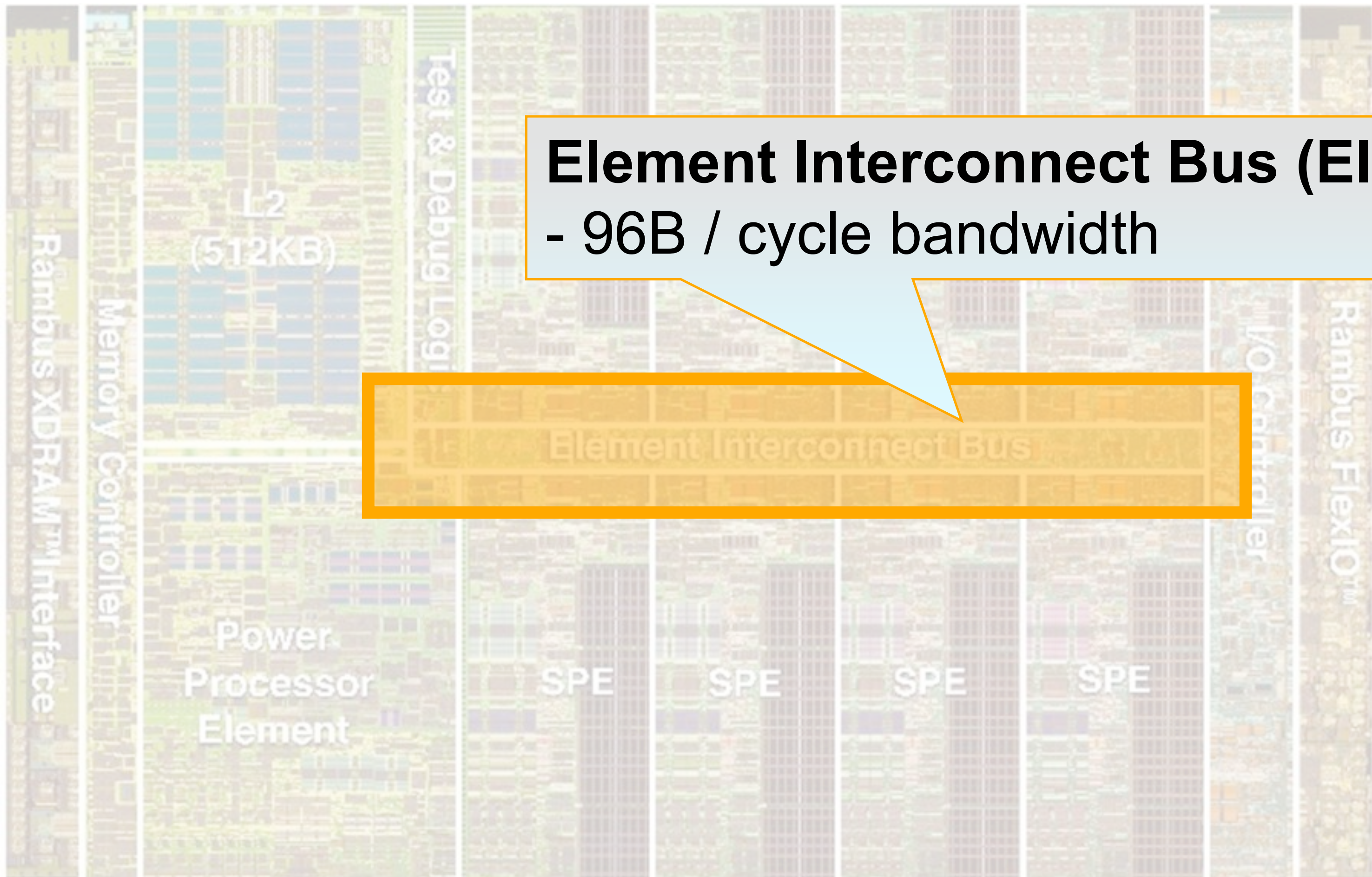
8 SPE cores

-128-bit SIMD instruction set

- Register file – 128x128-bit
- Local store – 256KB
- MFC
- Isolation mode

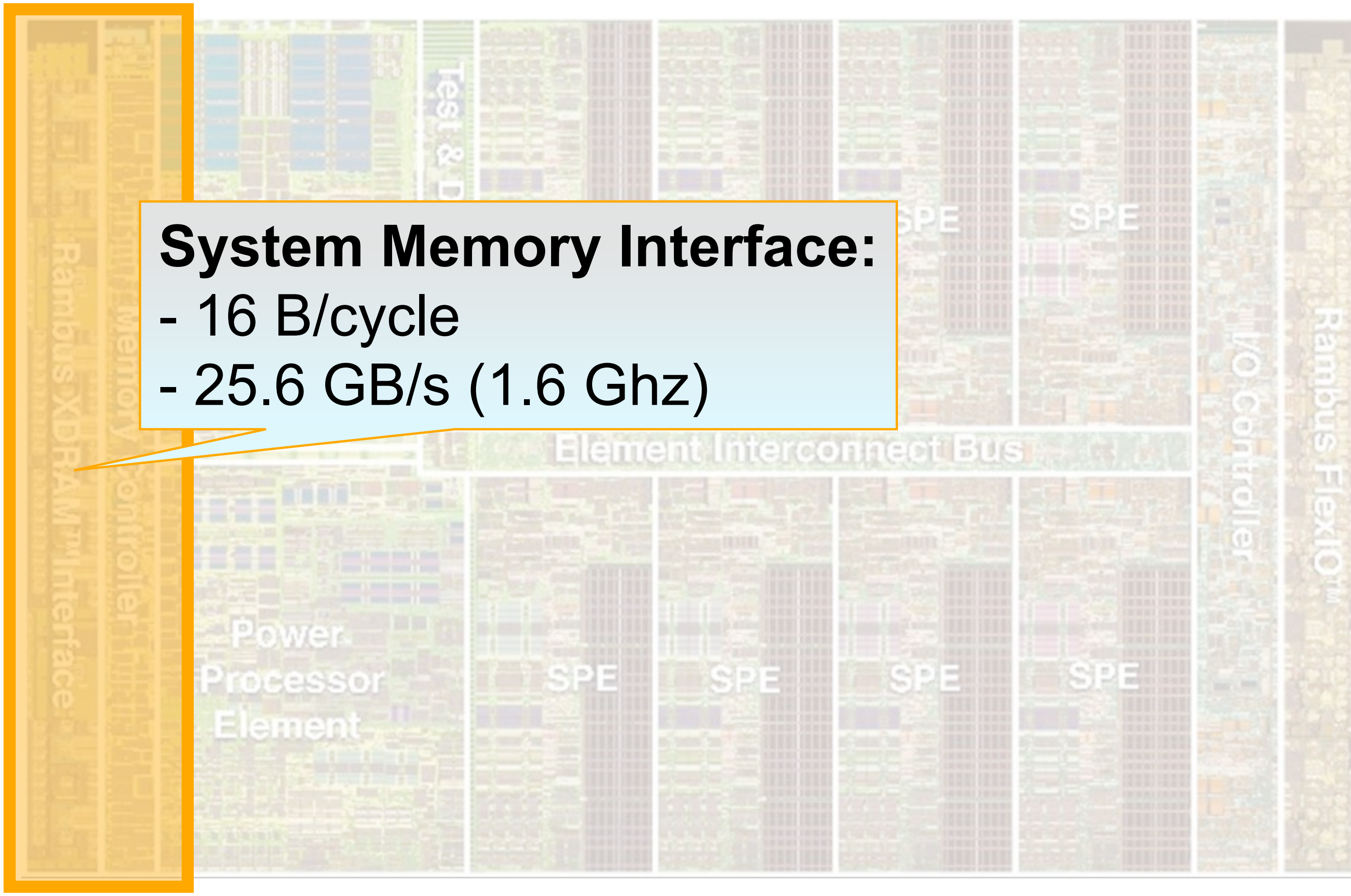


Element Interconnect Bus



Element Interconnect Bus (EIB)
- 96B / cycle bandwidth

System Memory Interface



System Memory Interface:
- 16 B/cycle
- 25.6 GB/s (1.6 Ghz)

Roadrunner lends itself to two general programming models

Host-centric model, e.g., SPaSM

Accelerator-centric model (inverted memory model), e.g., VPIC

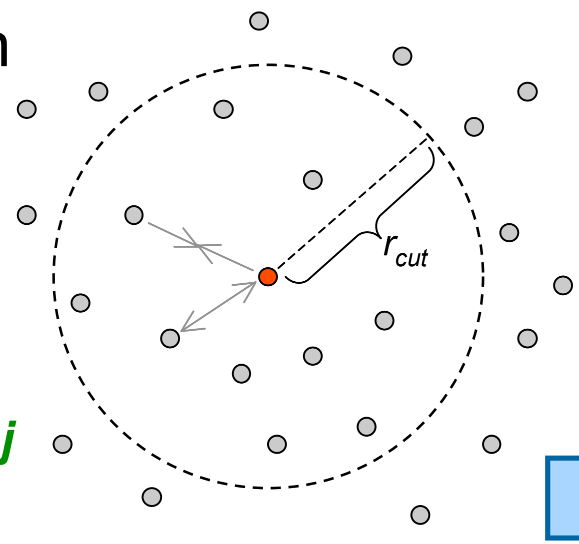
Roadrunner: Performance Considerations

Roadrunner exposes design concepts necessary for achieving performance on modern architectures

- Data motion – Overcoming memory latency and bandwidth limitations
  - DMA requests make data movement explicit and allow user to control when data are loaded
- Throughput - Use SIMD intrinsics
  - SPE vector processing units offer increased throughput
  - Static scheduling makes performance analysis/prediction more reliable
- Concurrency - Minimize thumb-twiddling
  - Support for data- and task-parallel programming models on SPEs
  - Problem decompositions for Roadrunner naturally adapt to homogeneous multicore architectures

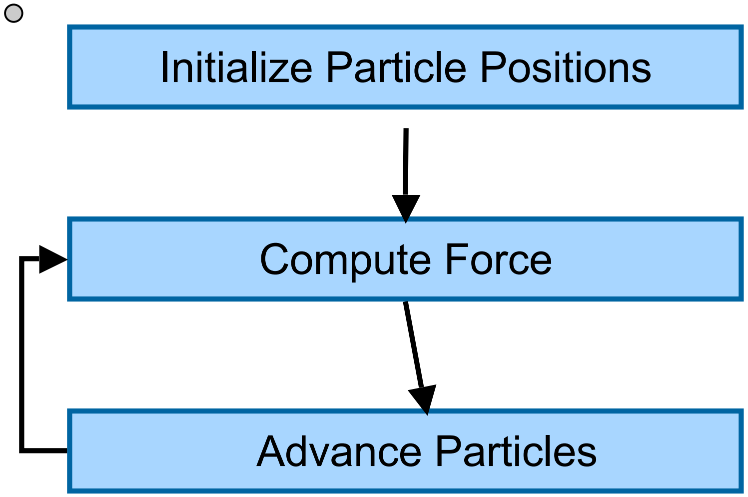
Data motion: For example, SPaSM Molecular Dynamics (MD) implementation

Force calculation



foreach particle *i*
    foreach neighbor *j*
        if $r_{ij} < r_{cut}$
            F_{ij} = interactions (*i,j*)
        end if
    end foreach
end foreach

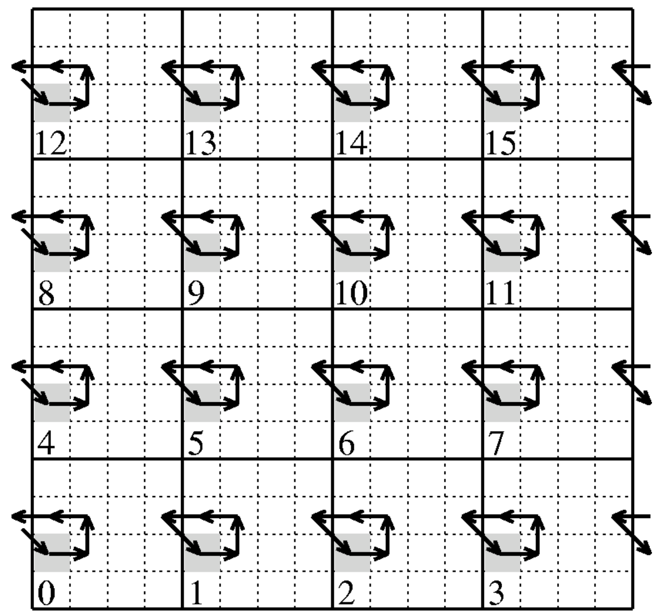
Time Iteration



Original SPaSM implementation

Designed when computation was more expensive than communication (e.g. Connection Machines)

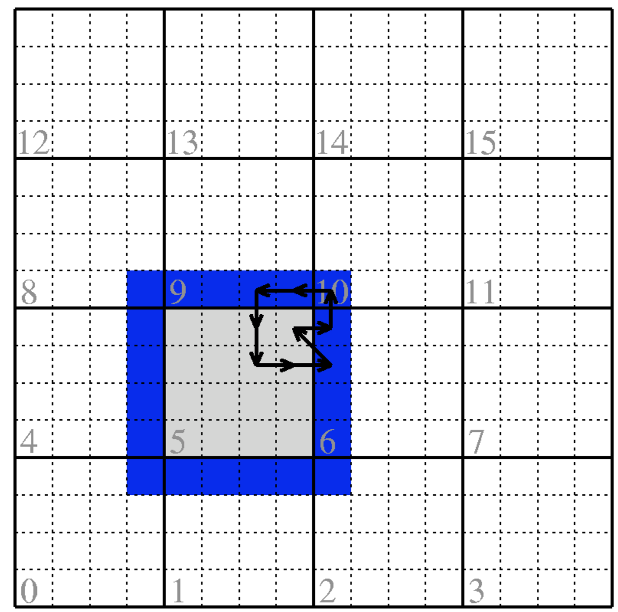
- **MPI processes advance through cells in lock-step**
- **Pair-wise force interactions are symmetric**
- **MPI send() and recv() calls used every time a remote neighbor is encountered**
- **Half neighbor list**



New SPaSM implementation: use full ghost-cell buffering to reduce communication

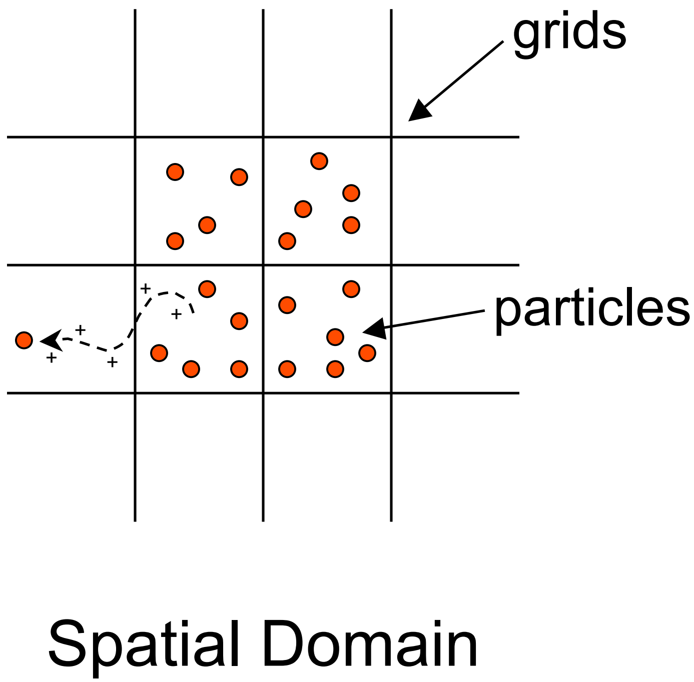
Reduces latency with fewer messages and allows for more straightforward data-level parallelism

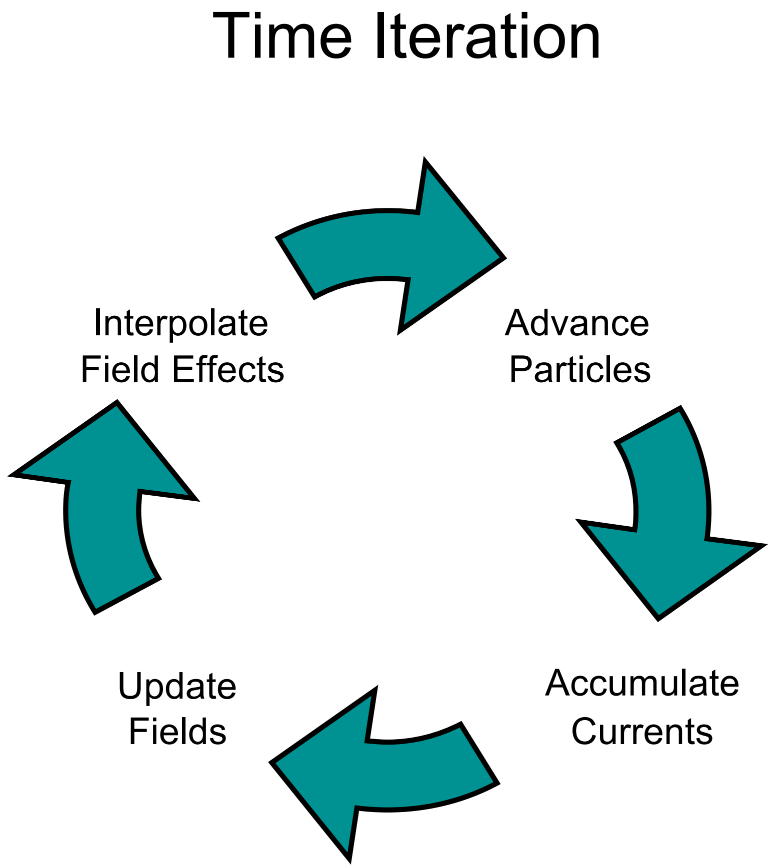
- **Blue ghost-cell region updated outside of particle interaction loop using MPI calls**
- **SPE threads can compute force interactions asynchronously without inter-node communication**
- **Current implementation uses full neighbor list**



VPIC design considerations for Roadrunner: a case study

VPIC is a Particle-In-Cell (PIC) kinetic plasma simulation method





Bowers et al. Phys. Plasmas 2007

VPIC is a flexible, general-purpose plasma physics code

- Plasmas are ionized gases with very complex dynamics.
- Understanding plasmas is important to many systems in basic science and national security, including:
  - **Thermonuclear burning plasma**
  - **Laser-plasma instabilities** for inertial confinement fusion experiments
  - **Magnetic fusion**
  - Diode modeling, radiography
  - Laser-particle accelerators
  - Space and astrophysics
- VPIC has been used to model all of these systems and more.

VPIC overview

- VPIC integrates the relativistic Maxwell-Boltzmann system in a linear background medium:

$$\partial_t f_s + c\gamma^{-1}\vec{u}\cdot\nabla f_s + \tfrac{q_s}{m_s c}\left(\vec{E} + c\gamma^{-1}\vec{u}\times\vec{B}\right)\cdot\nabla_u f_s = \left(\partial_t f_s\right)_{coll}$$

$$\partial_t \vec{E} = \varepsilon^{-1}\nabla\times\mu^{-1}\vec{B} - \varepsilon^{-1}\vec{J} - \varepsilon^{-1}\sigma\vec{E}$$

$$\partial_t \vec{B} = -\nabla\times\vec{E}$$

- Direct discretization of f_s is prohibitive; f_s is sampled by particles:

$$d_t\vec{r}_{s,n} = c\gamma_{s,n}^{-1}\vec{u}_{s,n} \qquad d_t\vec{u}_{s,n} = \tfrac{q_s}{m_s c}\left(\left.\vec{E}\right|_{\vec{r}_{s,n}} + c\gamma_{s,n}^{-1}\vec{u}_{s,n}\times\left.\vec{B}\right|_{\vec{r}_{s,n}}\right)$$

- Smooth *J* determined by the particles; *E*, *B* and *J* are sampled on a mesh and interpolated to and from particles

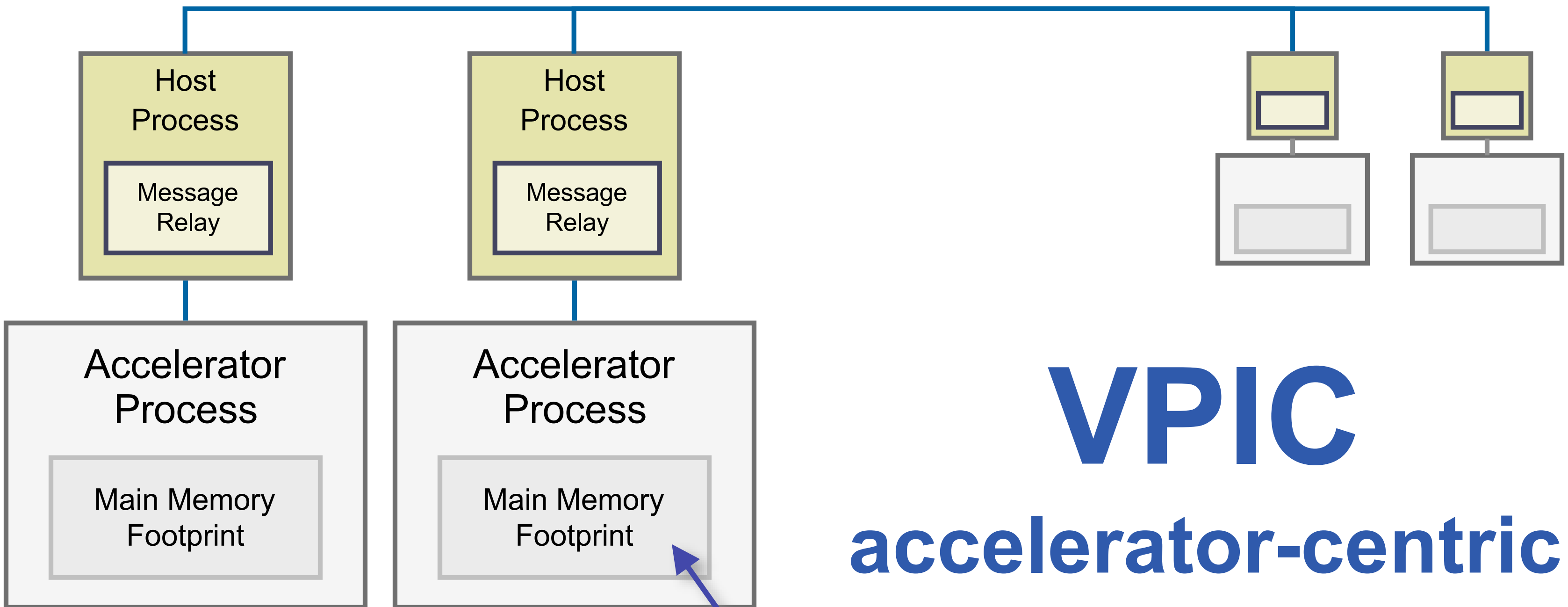
VPIC Design considerations for Roadrunner:

1. Data locality
2. Throughput
3. Concurrency

VPIC Design considerations for Roadrunner:

1. Data locality
2. Throughput
3. Concurrency

Data motion considerations forced our choice of programming model

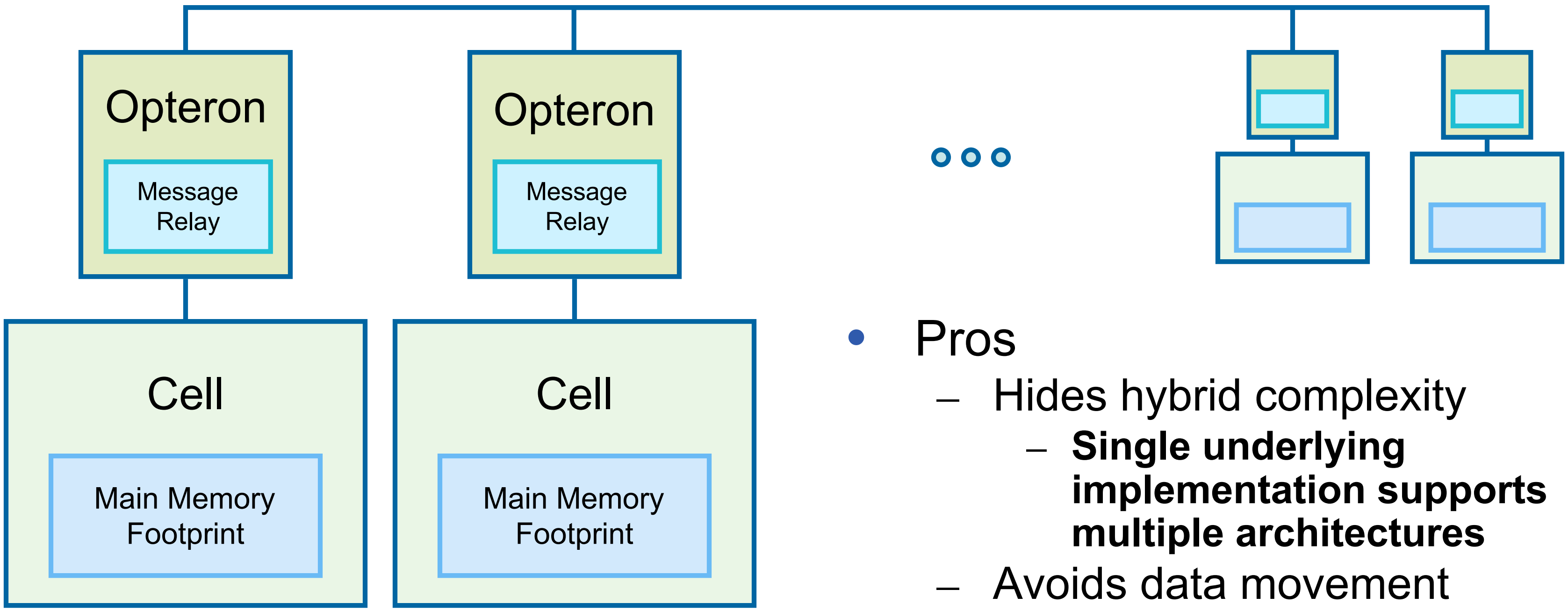


VPIC

accelerator-centric

VPIC has such a low compute/data ratio (common case: 246 ops/32 bytes), we locate the main memory as close to the SPE as possible!

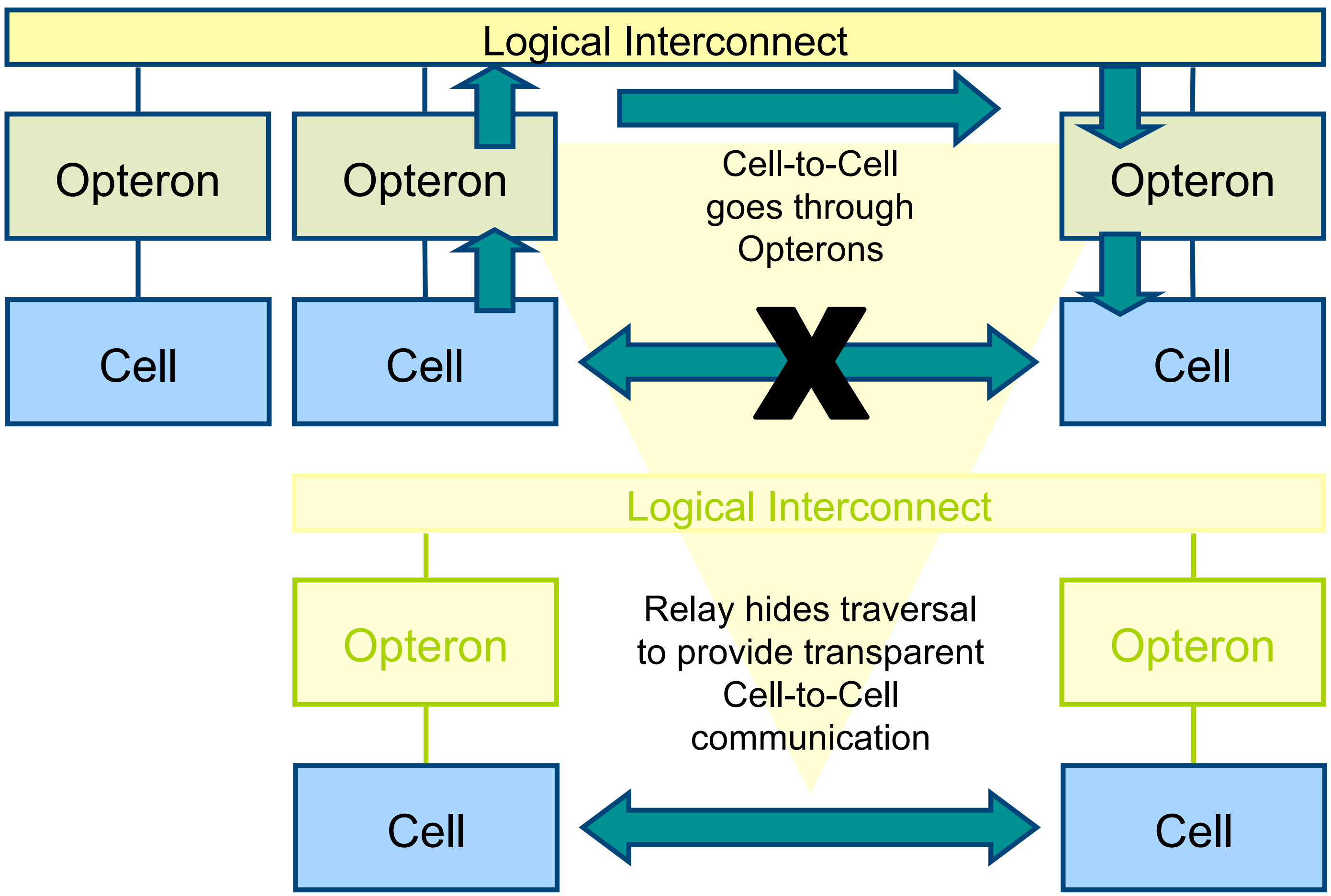
Accelerator-centric Programming Model



MPI traffic relayed through host

- Pros
  - Hides hybrid complexity
    - **Single underlying implementation supports multiple architectures**
  - Avoids data movement bottleneck over PCI-e communication path
- Cons
  - Requires full port to Cell
  - Potential PPE bottleneck

MP Relay: message relay layer



More on data motion: single pass processing and particle data layout

- We limit the number of times a particle is accessed during a time step (or else performance is limited by moving particle data to and from memory). Single pass processing achieves this:

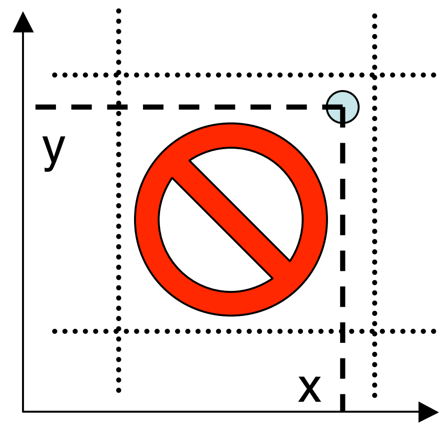
```
for each particle,
  interpolate E and B
  update u and compute movement
  update r and accumulate J
  if an exceptional boundary hit,
    save particle index and
      remaining movement
  end if
end for
```

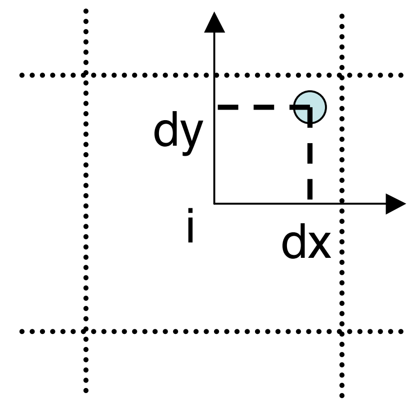
- To further minimize the cost of moving particle data, particle data is stored contiguously, memory aligned and organized for 4-vector SIMD
- The inner loop streams through particle data once using large aligned transfers under the hood—the ideal access pattern

```
typedef struct {
  float dx, dy, dz; int i; // Cell offset (on [-1,1]) and index
  float ux, uy, uz, q;     // Normalized momentum and charge
} particle_t;
```

Still more on data motion: VPIC was designed so that single precision would suffice

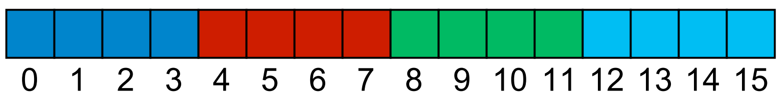
- Positions are given by the containing cell index and the offset from the cell center, normalized to the cell dimensions
- Various numerical "hygiene" techniques used
  - Divergence cleaning of E and B divergence errors
  - Radiation damping
- We are sensitive to roundoff (truncate gives about 10x the numerical heating as IEEE "round to even")



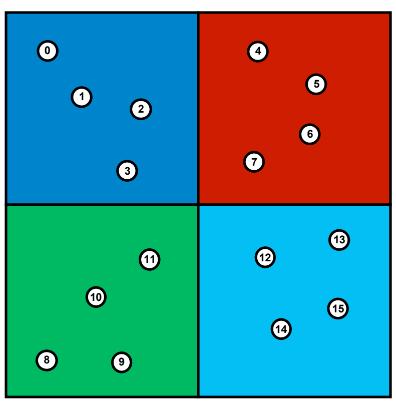


Yet more on data motion: maintaining locality in particle memory

Contiguous Memory



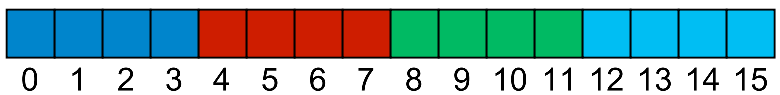
Compute Grid



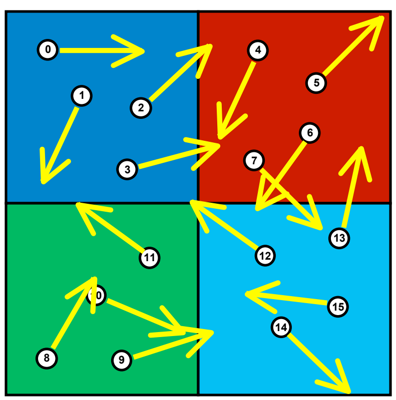
Naïve initial particle distribution by voxel places particle data spatially “close” in memory

Yet more on data motion: maintaining locality in particle memory

Contiguous Memory



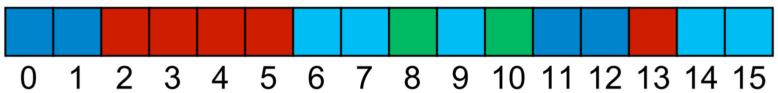
Compute Grid



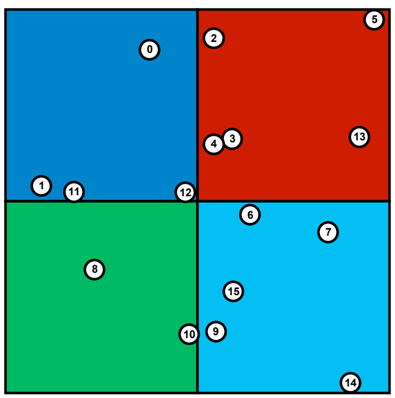
Advancing particles potentially moves them into new voxels

Yet more on data motion: maintaining locality in particle memory

Contiguous Memory



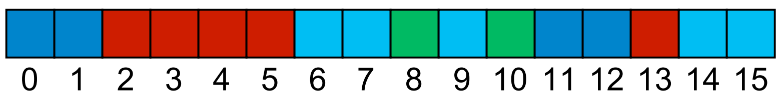
Compute Grid



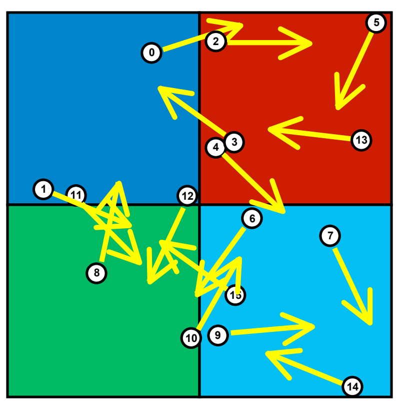
New particle positions interleave memory access with respect to voxels

Yet more on data motion: maintaining locality in particle memory

Contiguous Memory



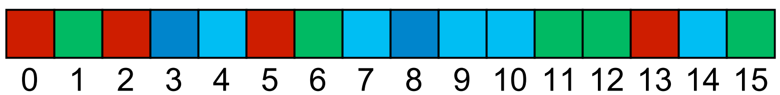
Compute Grid



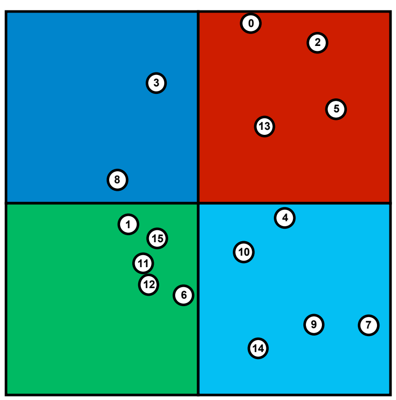
After several time iterations, particle data has lost spatial locality

Yet more on data motion: maintaining locality in particle memory

Contiguous Memory

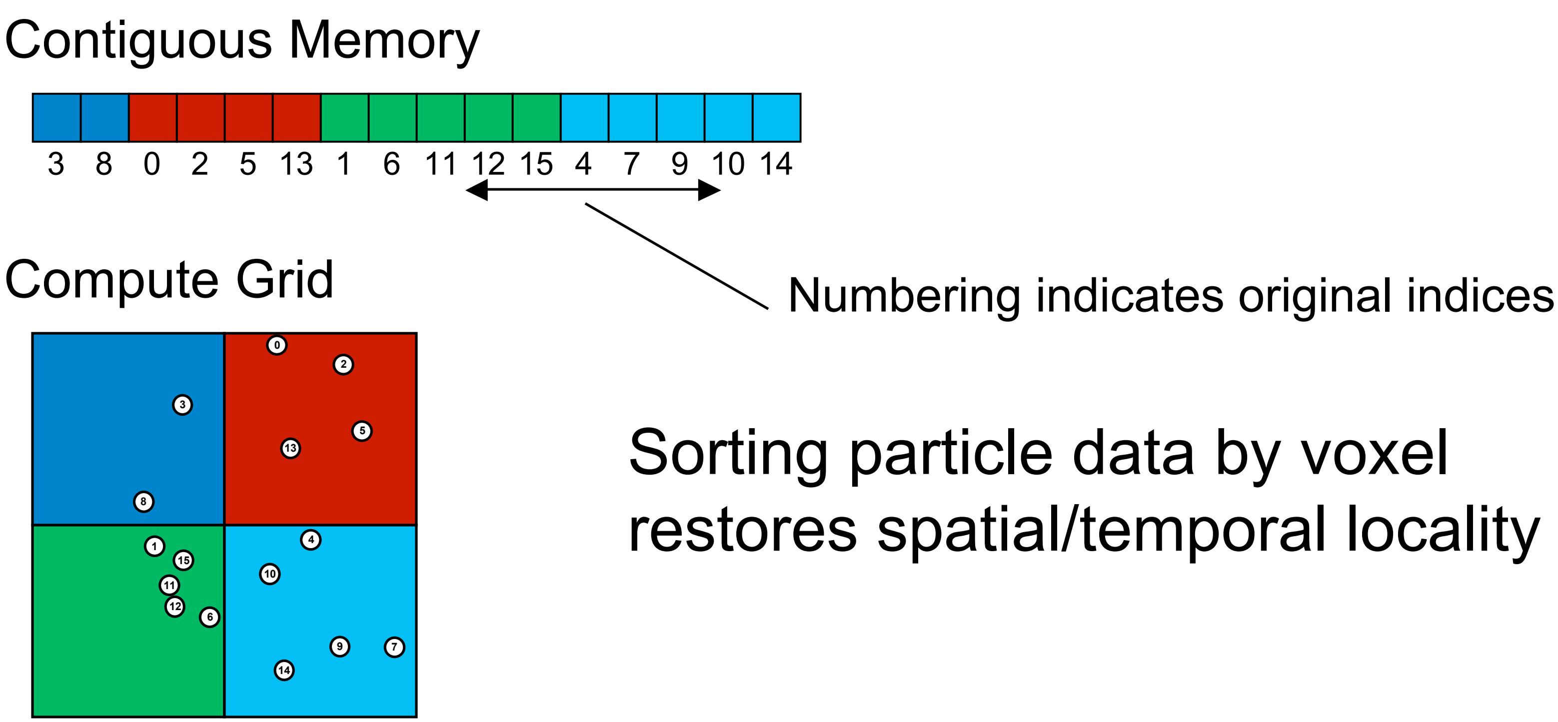


Compute Grid



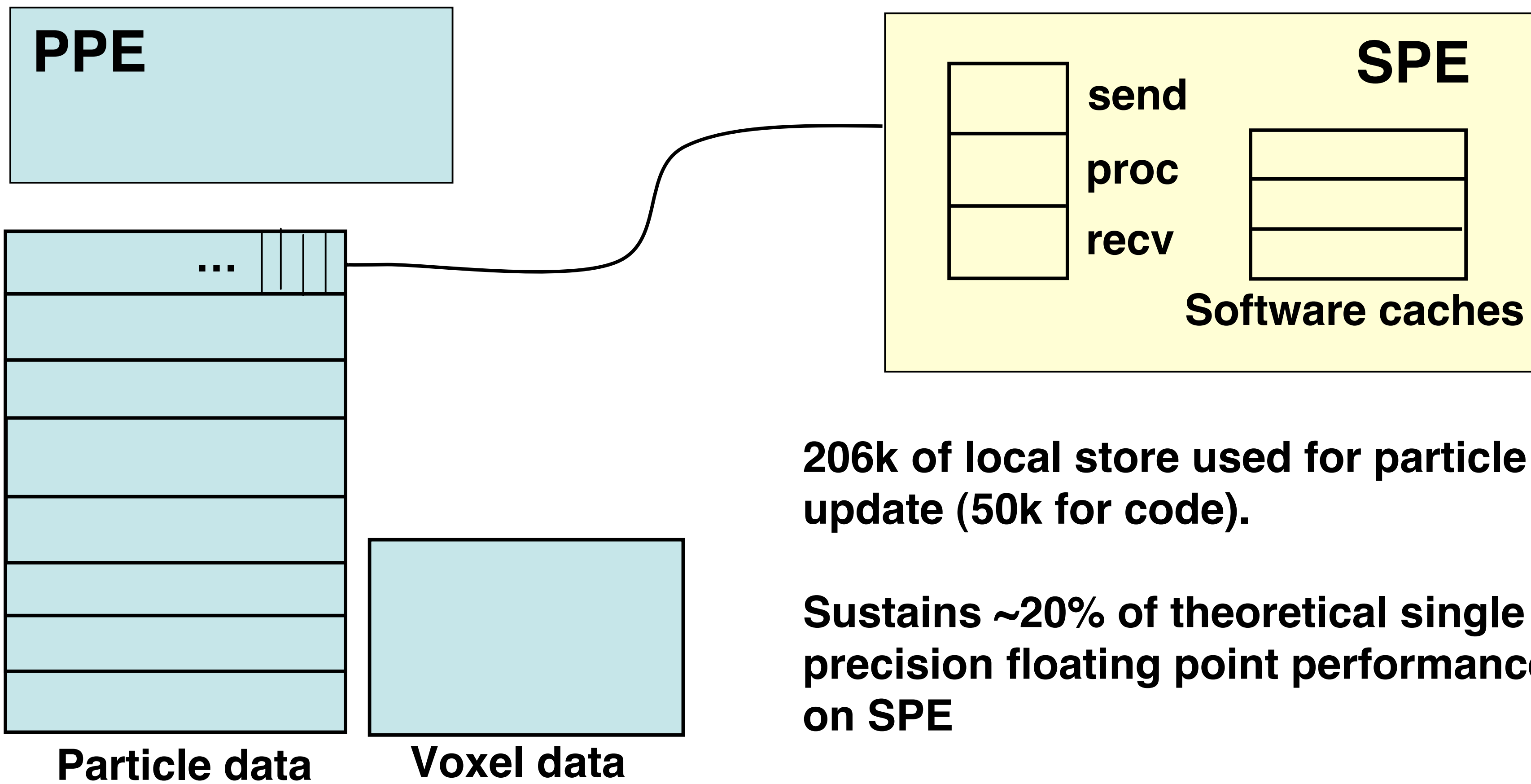
Loss of spatial locality in data access impacts temporal access of field data and hurts performance

Yet more on data motion: maintaining locality in particle memory



Sorting particle data by voxel restores spatial/temporal locality

VPIC particle advance uses (software) LRU caches and triple buffering



206k of local store used for particle update (50k for code).

Sustains ~20% of theoretical single precision floating point performance on SPE

VPIC Design considerations for Roadrunner:

1. Data locality
2. Throughput
3. Concurrency

Throughput: VPIC was designed around effective use of short-vector SIMD

```
// Interpolate ex for the next 4 particles
load_4x4_tr( interp_coeff[ i(0) ].QUAD( ex, dexdy, dexdz, d2exdydz ),
             interp_coeff[ i(1) ].QUAD( ex, dexdy, dexdz, d2exdydz ),
             interp_coeff[ i(2) ].QUAD( ex, dexdy, dexdz, d2exdydz ),
             interp_coeff[ i(3) ].QUAD( ex, dexdy, dexdz, d2exdydz ),
             ex, dexdy, dexdz, d2exdydz );
ex = (ex + dy*dexdy) + dz*(dexdz + dy*d2exdydz);
```

- Programming languages (e.g. C, FORTRAN) are not expressive enough (e.g. data alignment restrictions) to allow compilers to use 4-vector SIMD in operations as complex as those in VPIC
- VPIC has a language extension that allows C-style portable 4-vector SIMD code to be written and converted automatically to high performance 4-vector SIMD instructions on a wide variety of platforms. A similar approach was used in Bowers *et al* 2006
- First cut of migration of particle push from SSE to Cell SIMD took 1 day.

VPIC Design considerations for Roadrunner:

1. Data locality
2. Throughput
3. Concurrency

The core VPIC algorithm avoids MPI collectives and ensures a high degree of concurrency

- In vacuum, the field advance reduces to a FDTD method and the simulation must satisfy the Courant condition:

$$\left(\frac{c\delta_t}{\delta_x}\right)^2+\left(\frac{c\delta_t}{\delta_y}\right)^2+\left(\frac{c\delta_t}{\delta_z}\right)^2<1$$

Finite speed of light implies locality in field solve

- VPIC employs a so-called "charge conserving" scheme to avoid a Poisson (elliptic) solve:

$$\nabla\cdot\vec{J} = -\frac{\partial\rho}{\partial t}$$

$$-4\pi(\nabla\cdot\vec{J}) + c\underbrace{\nabla\cdot\nabla\times\vec{B}}_{=0} = 4\pi\frac{\partial\rho}{\partial t}$$

$$\nabla\cdot\underbrace{\left[c\,\nabla\times\vec{B} - 4\pi\vec{J}\right]}_{=\frac{\partial\vec{E}}{\partial t}} = 4\pi\frac{\partial\rho}{\partial t}$$

Apply $\int_0^t dt'$. Then, provided $\nabla\cdot\vec{E}=4\pi\rho$ initially, $\nabla\cdot\vec{E}=4\pi\rho$ thereafter.

Performance

Many applications were ported to Cell and hybrid and achieved significant speedup

Application	Type	Class	Cell Only (kernels)		Hybrid (Opteron+Cell)	
			CBE	eDP	CBE+IB	eDP+PCIe
SPaSM (10/07)	Science	full app	3x	4.5x	2.5x	>4x
SPaSM (now)			5x	7.5x	4x	>6x
VPIC	Science	full app	9x	9x	6x	>7x
Milagro	IC	full app	5x	6.5x	5x	>6x
Sweep3D	IC	kernel	5x	9x	5x	>5x

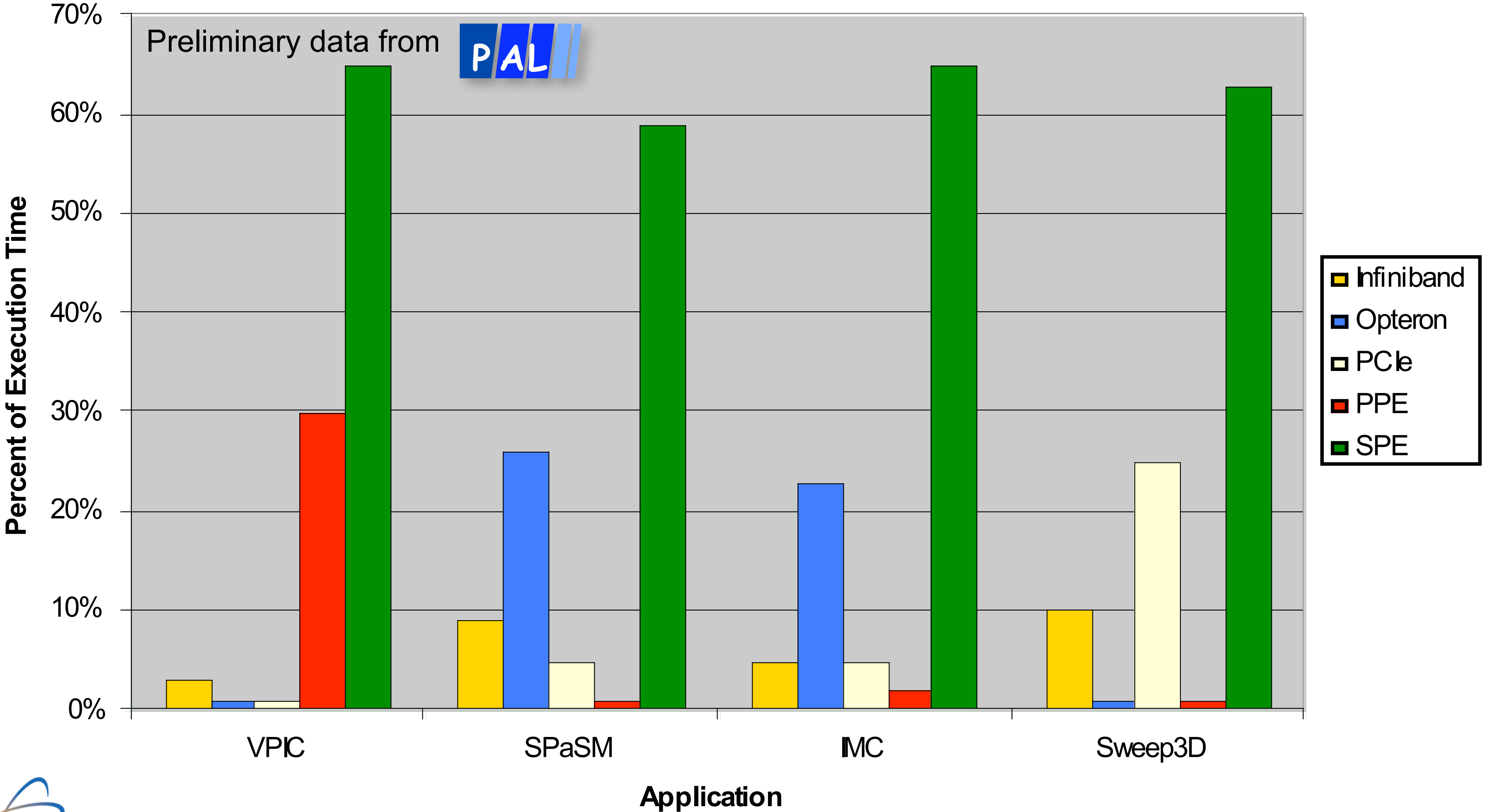
- all comparisons are to a single Opteron core
- parallel behavior unaffected, as will be shown in the scaling results
- first 3 columns are measured, last column is projected

These results were achieved with a relatively modest level of effort.

Code	Class	Language	Lines of code		FY07 FTEs
			Orig.	Modified	
VPIC	full app	C/C++	8.5k	10%	2
SPaSM	full app	C	34k	20%	2
Milagro	full app	C++	110k	30%	2 x 1
Sweep3D	kernel	C	3.5k	50%	2 x 1

- all staff started with little or no knowledge of Cell / hybrid programming
- 2 x 1 denotes separate efforts of roughly 1 FTE each
- most efforts also added code

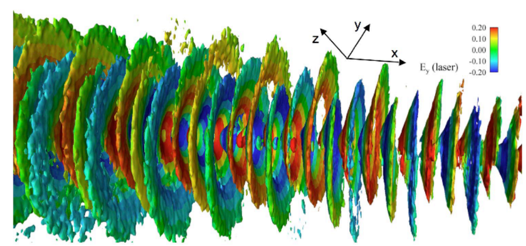
Roadrunner architecture is flexible - Applications are free to use hardware in most appropriate manner

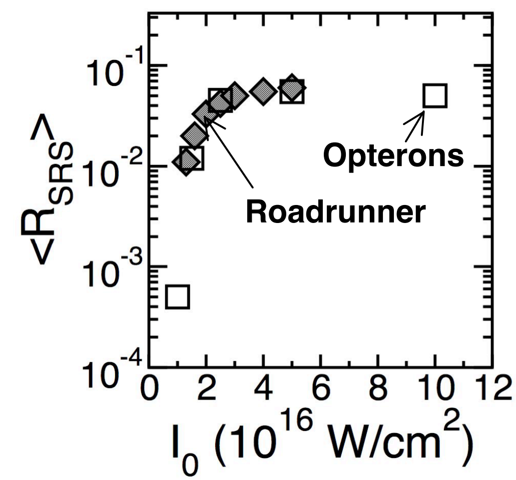


Roadrunner at IBM in Poughkeepsie - Highlights

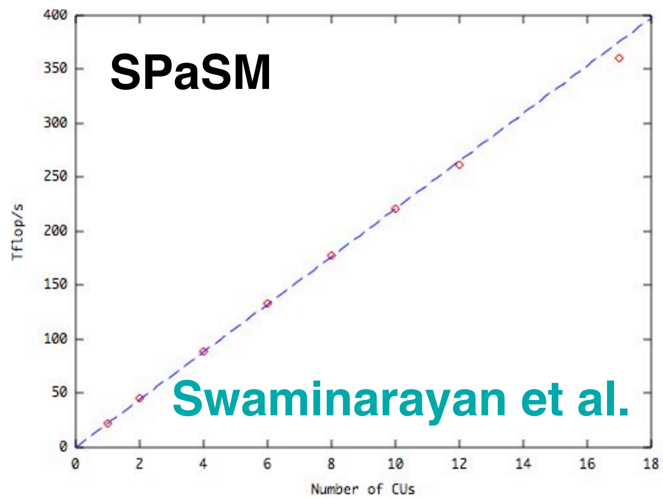
- Three LANL science applications (VPIC, SPaSM, and Petavision) were able to run in June in Poughkeepsie before system deployment at LANL
- All ran successfully on up to the entire machine (17 CU) and achieved predicted speedup.
- One application (VPIC) was able to run a series of science runs on up to 2 CU and achieved a 9x speedup over Opteron-only.
  - 9 of 10 runs completed; the 10th identified a DIMM failure on the machine.

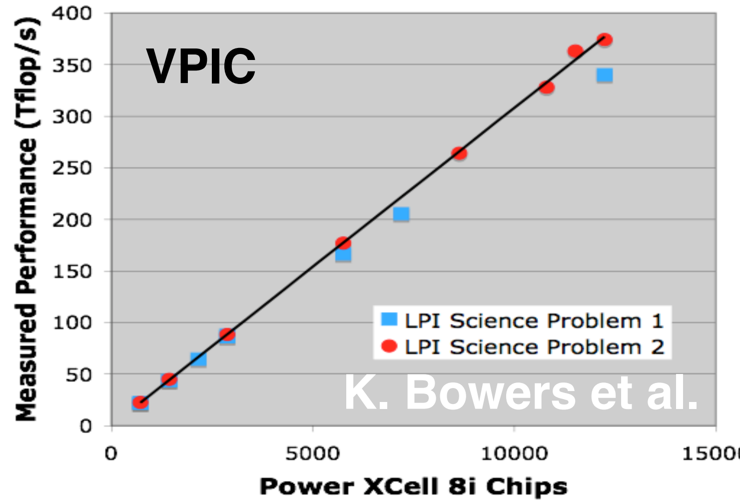
Electrostatic LPI fluctuations

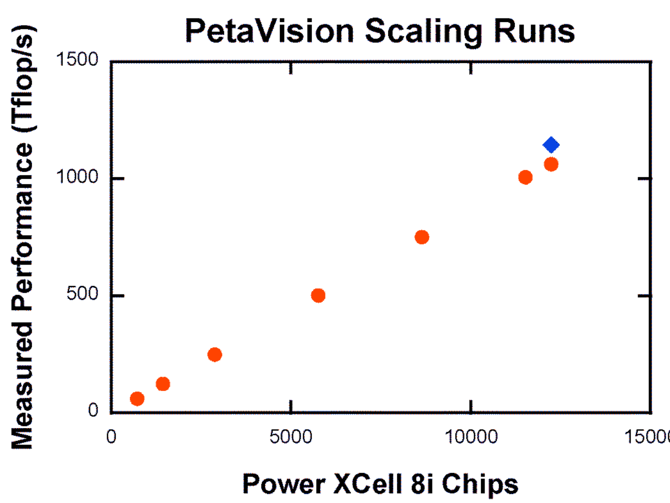




Excellent weak scalability was demonstrated by each application







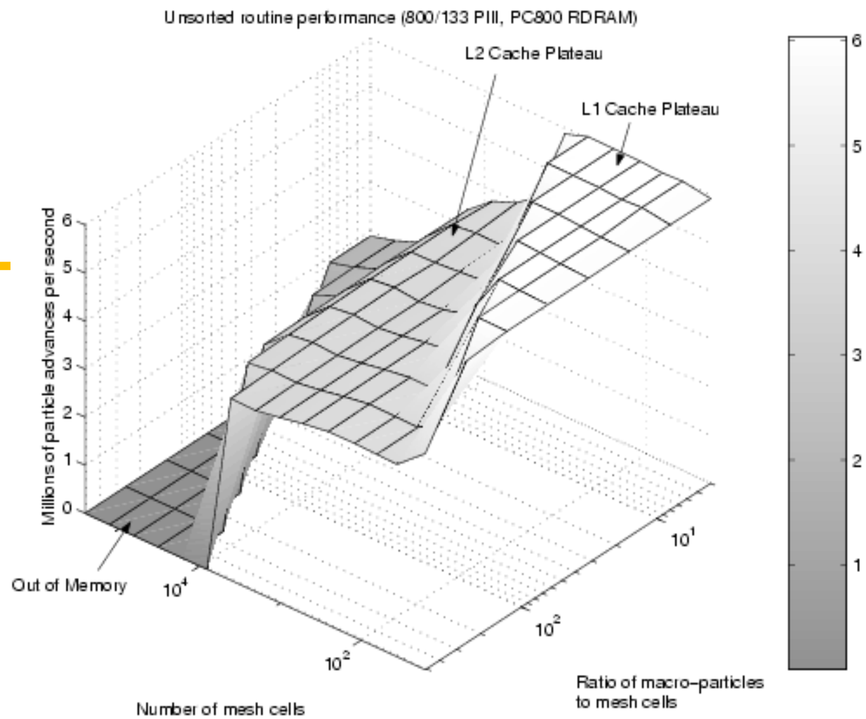
C. Rasmussen et al.

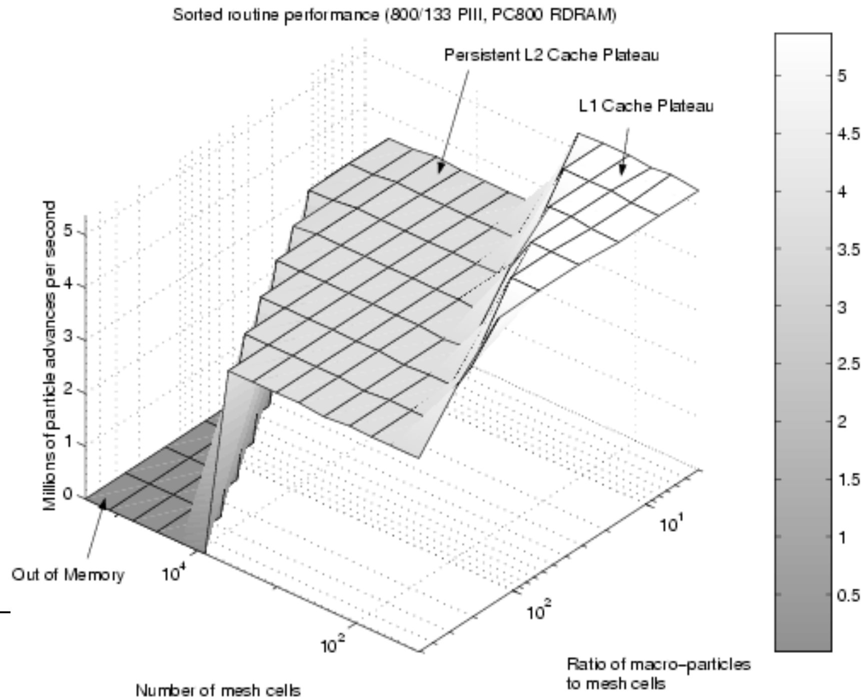
Conclusions

- Profound advances in supercomputing power are going to change the way we do science over the next decade.
- Tapping this potential requires that we rethink how we do supercomputing. We must optimize applications and algorithms for:
  - Data motion
  - Throughput
  - Concurrency
- Next-generation machines such as Roadrunner are an excellent place to develop algorithms; by designing for these platforms, one can “future-proof” applications for whatever the future brings.
- Several applications have already migrated successfully to hybrid platforms and have realized order-of-magnitude speedups over existing platforms. (See discussions in tomorrow’s meeting).

Particle sorting improves data locality

- Particles sorted periodically in *O(N)* by voxel index. They don't move far per step, so sorting is infrequent (tens to hundreds of time steps).
- We process particles (approximately) sequentially; field data loaded once from memory and cached.
- Improves performance on both homogeneous and hybrid platforms; accelerated sort being implemented





SPaSM Poughkeepsie Highlights

- Full 17 CU run achieved 361 TF
  - 26% of theoretical peak (double precision 1.376 PF)
  - 37 GF per Cell (36% of SPE peak)
  - Kernel operation achieves 45% of Cell theoretical peak
- Science runs (these will begin today)
  - Science runs will study the ejection of material from a copper crystal containing various surface imperfections and subjected to shock loading
  - 8 CUs for 8 hours each (at least two of these type)
  - 4 CUs for 48 hours (at least two of these types)
  - "Sweet spot" between 1-3 billion atoms per CU

PetaVision Highlights

- 500 million neuron simulation in visual cortex on 17CUs
  - Full run achieved sustained performance of 1.14 PF
    - 38% of theoretical max performance (single precision 3.0 PF)
    - 88 GF per Cell (43% of SPE peak)
  - Used simple neurons with Zucker connection weights
  - Excited by co-circular line segments
  - First large-scale calculation with Zucker weights and spiking neurons
- Next step: add a complex neuron layer with stored weights to add learning
- Ultimate goal of the project is synthetic cognition

Modest capability of Cell PPE: Get to play "Amdahl's Whack-a-mole"

- The Cell PPE, where VPIC lives, is a processor of modest performance.
- Highly optimized particle push means relative cost of other parts of algorithm creep up faster (particle sort, field advance, boundary handler).
- For very high performance, acceleration acquires more of an "all or nothing" character.